



Red Hat Process Automation Manager 7.8

Red Hat JBoss EAP クラスター環境への Red
Hat Process Automation Manager のインストール
および設定

ガイド

Red Hat Process Automation Manager 7.8 Red Hat JBoss EAP クラスター環境への Red Hat Process Automation Manager のインストールおよび設定

ガイド

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Installing_and_configuring_Red_Hat_Process_Automation_Manager_in_a_Red_Hat_JBoss_EAP_file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本書は、Red Hat JBoss Enterprise Application Platform 7.3 に Red Hat Process Automation Manager 7.8 クラスター環境を作成する方法を説明します。

目次

前書き	3
第1章 RED HAT PROCESS AUTOMATION MANAGER クラスター	4
第2章 開発 (オーサリング) 環境における RED HAT PROCESS AUTOMATION MANAGER クラスター	5
2.1. RED HAT DATA GRID のインストールおよび設定	5
2.2. AMQ BROKER のダウンロードおよび設定	6
2.3. NFS サーバーの設定	7
2.4. RED HAT JBOSS EAP 7.3 および RED HAT PROCESS AUTOMATION MANAGER のダウンロードおよび展開	8
2.5. クラスターでの BUSINESS CENTRAL の設定および実行	9
2.6. RED HAT PROCESS AUTOMATION MANAGER クラスターの検証	11
第3章 ランタイム環境における KIE SERVER クラスター	13
3.1. RED HAT JBOSS EAP 7.3 および KIE SERVER のダウンロードおよび展開	13
3.2. KIE SERVER 向けの RED HAT JBOSS EAP 7.3 クラスターの設定および実行	14
3.3. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーを使用した KIE SERVER のクラスタリング	17
第4章 SMART ROUTER のインストールおよび設定	19
4.1. SMART ROUTER を使用した KIE SERVER インスタンスの負荷分散	19
4.2. TLS 対応の SMART ROUTER の設定	22
4.3. エンドポイント認証用の SMART ROUTER の設定	22
4.4. SMART ROUTER 動作の設定	23
第5章 QUARTZ タイマーサービスの設定	25
第6章 関連資料	27
付録A バージョン情報	28

前書き

システムエンジニアは、Red Hat Process Automation Manager クラスター環境を作成して、開発環境およびランタイム環境に高可用性および負荷分散を提供できます。

前提条件

- 『[Red Hat Process Automation Manager インストールの計画](#)』の内容を確認している。

第1章 RED HAT PROCESS AUTOMATION MANAGER クラスター

2 台以上のコンピューターをクラスタリングすると、高可用性、コラボレーションの強化、負荷分散の利点があります。高可用性により、1 台のコンピューターで障害が発生したときにデータが損失する可能性を減らすことができます。その障害が発生したコンピューターにあるデータのコピーを提供することで、コンピューターに障害が発生したときに、別のコンピューターが不足を補います。障害が発生したコンピューターが再度オンラインになったら、クラスターに戻ります。負荷分散はクラスターのノード間でコンピューティング負荷を共有します。これにより、パフォーマンスが改善します。

Red Hat Process Automation Manager コンポーネントをクラスタリングを行う方法は複数あります。本書は、以下のシナリオにおけるクラスタリングの方法を説明します。

- [2章 開発\(オーサリング\) 環境における Red Hat Process Automation Manager クラスター](#)
- [3章 ランタイム環境における KIE Server クラスター](#)

第2章 開発 (オーサリング) 環境における RED HAT PROCESS AUTOMATION MANAGER クラスター



注記

Business Central の高可用性設定は現在、テクノロジープレビュー機能となっています。

開発者は、Red Hat Process Automation Manager を使用して、ユーザーの意思決定をサポートするルールとプロセスを作成できます。

Red Hat Process Automation Manager をクラスター開発環境として設定すると、高可用性の利点が得られます。クラスター環境では、開発者が `$node1` で作業していて、そのノードで障害が発生した場合に、この開発者が作業した内容はクラスターの別のノードに保存され、そちらで確認できます。

多くの開発環境には、ルールとプロセスを作成する Business Central と、このルールとプロセスをテストする 1 台以上の KIE Server があります。

Red Hat Process Automation Manager のクラスター開発環境を構築するには、以下のタスクを実行する必要があります。

- マシンに Red Hat JBoss EAP 7.3 with Red Hat Data Grid 7.3 を設定します。
- マシンに Java メッセージングサーバー (JMS) ブローカーである、AMQ Broker を設定します。
- マシンに NFS ファイルサーバーを設定します。
- Red Hat JBoss EAP 7.3 と Red Hat Process Automation Manager 7.8 をダウンロードして、各マシンにインストールします。各マシンはクラスターノードの 1 つとなります。
- クラスターノードごとに Business Central を設定して、クラスターの操作を開始します。

Red Hat Data Grid は Infinispan のオープンソースソフトウェアプロジェクトで構築されています。Red Hat Data Grid は、インデックス化の機能が含まれた、分散型インメモリーキー/値のデータストアで、大量のデータを素早くほぼリアルタイムに保存、検索、および解析できます。Red Hat Process Automation Manager のクラスター環境では、クラスターノード全体にわたる複雑な検索を効率的に実施できます。

JMS ブローカーは、ローカルでメッセージを受信して保存し、そのメッセージを受信者に転送するソフトウェアコンポーネントです。AMQ Broker を使用すると、アプリケーションがメッセージングプロバイダーと通信できます。また、メッセージ駆動型 Bean、Enterprise JavaBean、servlet などのコンポーネントがどのようにメッセージを送受信するかを指定します。

2.1. RED HAT DATA GRID のインストールおよび設定

クラスターノード全体でより効率的な検索を行うために、Red Hat Process Automation Manager のクラスター化環境に Red Hat Data Grid をインストールして設定します。

以下の手順を使用して、高可用性ではなく、簡素化された環境を別のマシンに設定します。

高度なインストールおよび設定オプション、ならびに Red Hat JBoss EAP の Red Hat Data Grid モジュールに関する情報は、[『Red Hat Data Grid User Guide』](#) を参照してください。



注記

Business Central と同じノードに Red Hat Data Grid はインストールしないでください。

前提条件

- Java 8.0 以降と互換性のある Java Virtual Machine (JVM) 環境がインストールされている。
- バックアップを作成してある Red Hat JBoss EAP システム (バージョン 7.3 またはそれ以降) が利用できる。Red Hat JBoss EAP システムのベースディレクトリーを **EAP_HOME** とします。
- Red Hat Process Automation Manager がインストールされ、設定されている。
- インストールを完了するために必要なユーザーパーミッションが付与されている。

手順

1. Red Hat カスタマーポータル [「Software Downloads」](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
 - **Product:** Data Grid
 - **Version:** 7.3
2. お使いのシステムの任意の場所に、Red Hat JBoss Data Grid 7.3.0 Server (**jboss-datagrid-7.3.0-1-server.zip**) のインストールファイルをダウンロードして展開します。展開したディレクトリーは、**JDG_HOME** となります。
3. Red Hat Data Grid を実行するには、**JDG_HOME/bin** に移動して以下のコマンドの1つを入力します。

- Linux または UNIX ベースのシステムの場合:

```
└─$ ./standalone.sh -c clustered.xml
```

- Windows の場合:

```
└─ standalone.bat -c clustered.xml
```



注記

Red Hat Data Grid を最新のバージョンに更新することを推奨します。詳細は、[『Red Hat Data Grid User Guide』](#) を参照してください。

2.2. AMQ BROKER のダウンロードおよび設定

Red hat AMQ Broker により、アプリケーションがメッセージングプロバイダーに接続できるようになります。また、メッセージ駆動型 Bean、Enterprise JavaBean、servlet などのコンポーネントがどのようにメッセージを送受信するかを指定します。

高可用性の Red Hat Process Automation Manager クラスター環境用の AMQ Broker を設定するには、[「Getting started with AMQ Broker」](#) を参照してください。

以下の手順を使用して、高可用性ではなく、簡素化された環境を設定します。

手順

- Red Hat カスタマーポータルの「[Software Downloads](#)」ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
 - Product: AMQ Broker
 - Version: 7.5.0
- Red Hat AMQ Broker 7.5.0(**amq-broker-7.5.0-bin.zip**) の横の **Download** をクリックします。
- amq-broker-7.5.0-bin.zip** ファイルを展開します。
- amq-broker-7.5.0-bin/amq-broker-7.5.0/bin** のディレクトリーに移動します。
- 以下のコマンドを入力します。以下のプレースホルダーを置き換えて、ブローカーおよびブローカーのユーザーを作成します。
 - <HOST> は、AMQ Broker をインストールしたサーバーの IP アドレスまたはホスト名に置き換えます。
 - <AMQ_USER> および <AMQ_PASSWORD> は、任意のユーザー名とパスワードの組み合わせに置き換えます。
 - <BROKER_NAME> は作成するブローカーの名前に置き換えます。

```
./artemis create --host <HOST> --user <AMQ_USER> --password <AMQ_PASSWORD>
--require-login <BROKER_NAME>
```

- AMQ Broker を実行するには、**amq-broker-7.5.0-bin/amq-broker-7.5.0/bin** ディレクトリーで以下のコマンドを入力します。

```
amq-broker-7.5.0/bin/<BROKER_NAME>/bin/artemis run
```

2.3. NFS サーバーの設定

Business Central クラスター環境には、共有ファイルシステムが必要で、その共有ファイルシステムに、各クラスターノードからアクセスできる必要がある。

NFS バージョン 4 サーバーをデプロイして設定しておく必要がある。

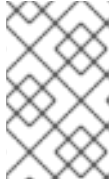
手順

- NFS バージョン 4 共有をエクスポートするようにサーバーを設定します。Red Hat Enterprise Linux での NFS 共有のエクスポートの方法については、『[ファイルシステムの管理](#)』の「[NFS 共有のエクスポート](#)」を参照してください。NFS サーバーの作成に関する情報は、『[RHEL 7 で NFS を設定する](#)』を参照してください。
- サーバーで、**/etc/exports** ファイルに以下の行を追加し、**rw,sync,no_root_squash** オプションを指定して **/opt/kie/data** 共有を作成します。

```
/opt/kie/data *(rw,sync,no_root_squash)
```

たとえば、**/opt/kie/data** は共有フォルダー、***** は NFS サーバーに接続可能な IP アドレス、**(rw,sync,no_root_squash)** は NFS に最小限必要なオプションを指します。例:

```
/opt/kie/data 192.268.1.0/24(rw,sync,no_root_squash)
```



注記

'/opt/kie/data' の代わりに別の共有名を使用できます。別の共有名を使用する場合には、Business Central を実行する全ノードの設定時に、この名前を使用する必要があります。

- 各クライアントノードで、既存のディレクトリーに共有フォルダーをマウントします。

```
# mount <SERVER_IP>:/opt/kie/data /opt/kie/data/niogit
```

- 以下のプロパティを **standalone-full-ha.xml** ファイルに追加し、**.niogit** および **maven-repository** ディレクトリーを nfs 共有フォルダーとしてバインドします。

```
<property name="org.uberfire.nio.git.dir" value="/opt/kie/data/niogit"/>
<property name="org.guvnor.m2repo.dir" value="/opt/kie/data/maven-repository"/>
```

2.4. RED HAT JBOSS EAP 7.3 および RED HAT PROCESS AUTOMATION MANAGER のダウンロードおよび展開

クラスターの各ノードで Red Hat JBoss EAP 7.3 および Red Hat Process Automation Manager 7.8 をダウンロードして、インストールします。

手順

- クラスターの各ノードに Red Hat JBoss EAP 7.3 をインストールします。
 - Red Hat カスタマーポータル [「Software Downloads」](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
 - **Product: Enterprise Application Platform**
 - **Version: 7.3**
 - Red Hat JBoss Enterprise Application Platform 7.3.0の横にある **Download** をクリックします。 (**JBEAP-7.3.0/jboss-eap-7.3.0.zip**)。
- jboss-eap-7.3.0.zip** ファイルを展開します。以下の例では、**EAP_HOME** を **jboss-eap-7.3/jboss-eap-7.3** ディレクトリーとします。
- 最新の Red Hat JBoss EAP パッチが利用できる場合には、ダウンロードして適用します。
- クラスターの各ノードで Red Hat Process Automation Manager をダウンロードします。
 - Red Hat カスタマーポータル [「Software Downloads」](#) ページに移動し、ドロップダウンオプションから製品およびバージョンを選択します。
 - **Product: Process Automation Manager**
 - **バージョン: 7.8**

- b. **Red Hat Process Automation Manager 7.8.0 Business Central Deployable for Red Hat JBoss EAP 7 (rhpam-7.8.0-business-central-eap7-deployable.zip)** をダウンロードします。
5. **rhpam-7.8.0-business-central-eap7-deployable.zip** ファイルを一時ディレクトリーに展開します。以下のコマンドでは、このディレクトリーを **TEMP_DIR** とします。
6. **TEMP_DIR/rhpam-7.8.0-business-central-eap7-deployable/jboss-eap-7.3** の内容を **EAP_HOME** にコピーします。
7. 最新の Red Hat Process Automation Manager のパッチが利用できる場合には、ダウンロードして適用します。
8. **EAP_HOME/bin** ディレクトリーに移動します。
9. Business Central へのログインに使用する、**admin** ロール持つユーザーを作成します。以下のコマンドの **<username>** および **<password>** を、作成するユーザーとそのパスワードに置き換えます。

```
$ ./add-user.sh -a --user <USERNAME> --password <PASSWORD> --role admin,rest-all
```



注記

必ず、既存のユーザー、ロール、またはグループとは異なるユーザー名を指定してください。たとえば、**admin** という名前のユーザーは作成しないでください。

パスワードは 8 文字以上で、数字と、英数字以外の文字をそれぞれ 1 文字以上使用する必要があります。ただし & の文字は使用できません。

高可用性環境では、LDAP または RH-SSO を使用する必要があります。詳細は、『[Red Hat Single Sign-On サーバー管理ガイド](#)』を参照してください。

10. KIE Server へのログインに使用する **kie-server** ロールのユーザーを作成します。

```
$ ./add-user.sh -a --user <USERNAME> --password <PASSWORD> --role kie-server
```

11. ユーザー名とパスワードを書き留めておきます。

2.5. クラスターでの BUSINESS CENTRAL の設定および実行

Red Hat JBoss EAP と Business Central のインストール後に、Red Hat Data Grid と AMQ Broker を使用してクラスターを設定できます。クラスターの各ノードで以下の手順を実行します。



注記

この手順では、基本的なクラスター設定を説明します。より詳細な設定は、『[Red Hat JBoss EAP 7.3 設定ガイド](#)』を参照してください。

前提条件

- 『[Red Hat Data Grid のインストールおよび設定](#)』の説明通りに、Red Hat Data Grid 7.3 がインストールされている。

- 「AMQ Broker のダウンロードおよび設定」の説明通りに AMQ Broker がインストールされ、設定されている。
- 「Red Hat JBoss EAP 7.3 および Red Hat Process Automation Manager のダウンロードおよび展開」の説明通りに、クラスターの各ノードに Red Hat JBoss EAP および Red Hat Process Automation Manager がインストールされている。
- 「NFS サーバーの設定」で記載されているように、パーティションをマウントした NFS サーバーが利用できる。

手順

1. NFS で共有されているディレクトリーを `/data` としてマウントします。Root ユーザーで以下のコマンドを入力します。

```
mkdir /data
mount <NFS_SERVER_IP>:<DATA_SHARE> /data
```

`<NFS_SERVER_IP>` を、NFS サーバーマシンの IP アドレスまたはホスト名に置き換えてください。`<DATA_SHARE>` を、設定した共有名 (例: `/opt/kie/data`) に置き換えます。

2. テキストエディターで `EAP_HOME/standalone/configuration/standalone-full.xml` ファイルを開きます。
3. `<system-properties>` 要素でプロパティを編集または追加し、以下のプレースホルターを置き換えます。
 - `<AMQ_USER>` および `<AMQ_PASSWORD>` は、AMQ Broker の作成時に定義した認証情報に置き換えます。
 - `<AMQ_BROKER_IP_ADDRESS>` は AMQ Broker の IP アドレスに置き換えます。
 - `<INFINISPAN_NODE_IP>` は、Red Hat Data Grid のインストール先の IP アドレスに置き換えます。

```
<system-properties>
  <property name="appformer-jms-connection-mode" value="REMOTE"/>
  <property name="appformer-jms-username" value="<AMQ_USER>"/>
  <property name="appformer-jms-password" value="<AMQ_USER_PASSWORD>"/>
  <property name="appformer-jms-url"
    value="tcp://<AMQ_BROKER_IP_ADDRESS>:61616?
    ha=true&retryInterval=1000&retryIntervalMultiplier=1.0&reconnectAttempts=
    -1"/>
  <property name="org.appformer.ext.metadata.infinispan.port"
    value="11222"/>
  <property name="org.appformer.ext.metadata.infinispan.host"
    value="<INFINISPAN_NODE_IP>"/>
  <property name="org.appformer.ext.metadata.infinispan.realm"
    value="ApplicationRealm"/>
  <property name="org.appformer.ext.metadata.infinispan.cluster"
    value="kie-cluster"/>
  <property name="org.appformer.ext.metadata.index"
    value="infinispan"/>
  <property name="org.uberfire.nio.git.dir"
    value="/data"/>
```

```
<property name="es.set.netty.runtime.available.processors"
  value="false"/>
</system-properties>
```

- オプション: Red Hat Data Grid のデプロイメントで認証が必要な場合は、**<system-properties>** 要素でプロパティを編集または追加し、以下のプレースホルターを置き換えます。

- **<SERVER_NAME>** は、Red Hat Data Grid のサーバー設定で指定したサーバー名に置き換えます。
- **<SASL_QOP>** は、Red Hat Data Grid サーバー設定の auth、auth-int、および auth-conf の値に置き換えます。

```
<property name="org.appformer.ext.metadata.infinispan.server.name"
  value="<SERVER_NAME>"/>
<property name="org.appformer.ext.metadata.infinispan.sasl.qop"
  value="<SASL_QOP>"/>
<property name="org.appformer.ext.metadata.infinispan.username"
  value=""/>
<property name="org.appformer.ext.metadata.infinispan.password"
  value=""/>
```

5. **standalone-full.xml** ファイルを保存します。
6. クラスタを起動するには **EAP_HOME/bin** に移動して、以下のコマンドの1つを入力します。
 - Linux または UNIX ベースのシステムの場合:

```
$ ./standalone.sh -c standalone-full.xml
```

- Windows の場合:

```
standalone.bat -c standalone-full.xml
```

2.6. RED HAT PROCESS AUTOMATION MANAGER クラスタの検証

Red Hat Process Automation Manager を設定したら、アセットを作成してシステムが機能していることを検証します。

手順

1. Web ブラウザーで、**<node-IP-address>:8080/business-central** を入力します。 **<node-IP-address>** は特定のノードの IP アドレスに置き換えます。
2. インストール時に作成した **admin** ユーザーの認証情報を入力します。Business Central ホームページが表示されます。
3. **Menu** → **Design** → **Projects** の順に選択します。
4. **MySpace** スペースを開きます。
5. **Try Samples** → **Mortgages** → **OK** の順にクリックします。 **Assets** ウィンドウが表示されます。

6. **Add Asset** → **Data Object** をクリックします。
7. **Data Object** フィールドに **MyDataObject** と入力し、**OK** をクリックします。
8. **Spaces** → **mySpace** → **Mortgages** の順にクリックし、アセットリストに **MyDataObject** があることを確認します。
9. Web ブラウザーに以下の URL を入力します。<node_IP_address> には、クラスターの別のノードのアドレスに置き換えます。
http://<node_IP_address>:8080/business-central
10. **MyDataObject** アセットを作成した最初のノードの Business Central にログインするときに使用した認証情報と同じものを入力します。
11. **Menu** → **Design** → **Projects** の順に選択します。
12. **MySpace** スペースを開きます。
13. **Mortgages** プロジェクトを選択します。
14. **MyDataObject** がアセットリストにあることを確認します。
15. **Mortgages** プロジェクトを選択します。

第3章 ランタイム環境における KIE SERVER クラスター

ランタイム環境では、KIE Server は、ビジネス上の決定をサポートするルールおよびプロセスが含まれるサービスを実行します。KIE Server ランタイム環境をクラスタリングする主な利点は負荷分散です。クラスターのノードの1つでアクティビティが増えると、そのアクティビティはクラスターの残りのノードと共有されるため、パフォーマンスが改善します。

KIE Server のクラスター化ランタイム環境を作成するには、Red Hat JBoss EAP 7.3 および KIE Server をダウンロードして展開します。次に、ドメインモードクラスターに Red Hat JBoss EAP 7.3 を設定し、クラスターを起動し、各クラスターノードに KIE Server をインストールします。

任意で、ヘッドレス Process Automation Manager コントローラーおよび Smart Router をクラスター化できます。

3.1. RED HAT JBOSS EAP 7.3 および KIE SERVER のダウンロードおよび展開

本セクションの手順を行い、Red Hat JBoss EAP 7.3 をダウンロードしてインストールし、クラスター環境にインストールするために KIE Server をダウンロードして再パッケージ化します。

手順

1. クラスターの各ノードに Red Hat JBoss EAP 7.3 をインストールします。
 - a. Red Hat カスタマーポータル [「Software Downloads」](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
 - **Product: Red Hat JBoss EAP**
 - **Version: 7.3**
 - b. Red Hat JBoss Enterprise Application Platform 7.3.0 (**jboss-eap-7.3.0.zip**) の横にある **Download** をクリックします。
2. **jboss-eap-7.3.0.zip** ファイルを展開します。 **jboss-eap-7.3/jboss-eap-7.3** ディレクトリーは **EAP_HOME** とします。
3. 最新の Red Hat JBoss EAP パッチが利用できる場合には、ダウンロードして適用します。
4. KIE Server のダウンロード:
 - a. Red Hat カスタマーポータル [「Software Downloads」](#) ページに移動し、ドロップダウンオプションから製品およびバージョンを選択します。
 - **Product: Process Automation Manager**
 - **バージョン: 7.8**
 - b. Red Hat Process Automation Manager 7.8.0 KIE Server for All Supported EE8 Containers (**rhcam-7.8.0-kie-server-ee8.zip**) をダウンロードします。
5. **rhcam-7.8.0-kie-server-ee8.zip** アーカイブを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP_DIR** とします。
6. **TEMP_DIR/rhcam-7.8.0-kie-server-ee8/rhcam-7.8.0-kie-server-ee8/kie-server.war** ディレクトリーを **EAP_HOME/standalone/deployments/** にコピーします。



警告

コピーする Red Hat Process Automation Manager デプロイメントの名前が、Red Hat JBoss EAP インスタンスの既存デプロイメントと競合しないことを確認します。

7. **TEMP_DIR/rhpam-7.8.0-kie-server-ee8/rhpam-7.8.0-kie-server-ee8/SecurityPolicy/** の内容を **EAP_HOME/bin** にコピーします。ファイルの上書きを確認するメッセージが表示されたら、**Replace** をクリックします。
8. **EAP_HOME/standalone/deployments/** ディレクトリーに、**kie-server.war.dodeploy** という名前で空のファイルを作成します。このファイルにより、サーバーが起動すると KIE Server が自動的にデプロイされます。
9. 最新の Red Hat Process Automation Manager のパッチが利用できる場合には、ダウンロードして適用します。
10. **EAP_HOME/bin** ディレクトリーに移動します。
11. KIE Server へのログインに使用する **kie-server** ロールのユーザーを作成します。

```
$ ./add-user.sh -a --user <USERNAME> --password <PASSWORD> --role kie-server
```

12. ユーザー名とパスワードを書き留めておきます。

3.2. KIE SERVER 向けの RED HAT JBOSS EAP 7.3 クラスターの設定および実行

KIE Server 向けに Red Hat JBoss EAP クラスターを設定して、起動します。

手順

1. JDBC ドライバーを、このクラスターの一部となる Red Hat JBoss EAP の全インスタンスにインストールします。詳細は、『Red Hat JBoss EAP 7.3 設定ガイド』の「[JDBC ドライバー](#)」セクションを参照してください。
2. テキストエディターで **EAP_HOME/standalone/configuration/standalone-full.xml** ファイルを開きます。
3. **data-stores** プロパティーと、その上の **timer-service thread-pool-name** を編集します。
 - **datasource-jndi-name** は、この手順の最初で指定したデータベースの JNDI 名です。
 - **partition** プロパティーの値にはあらゆる名前を入力できますが、ノードには、同じパーティション名を持つその他のノードのタイマーのみが表示されます。パーティション名を割り当てて、パーティションでノードをグループ分けすると、大規模なクラスターを複数の小規模クラスターに分割できます。これによりパフォーマンスが改善します。これを行うと、パフォーマンスが向上します。たとえば、1つのクラスターにノードが100個あり、

100 個のノードがすべて同じタイマーを実行して更新する代わりに、ノードを 5 個ずつに分割して 20 個のクラスターを作成し、各クラスターに異なるパーティション名を指定することもできます。

- **default-data-store** 属性値を **ejb_timer_ds** に置き換えます。
- **refresh-interval** の値をミリ秒で設定して、EJB タイマーがデータベースに接続して同期し、処理するタスクをロードする頻度を指定します。

```
<timer-service thread-pool-name="default" default-data-store="ejb_timer_ds">
<data-stores>
  <database-data-store name="ejb_timer_ds" datasource-jndi-
name="java:jboss/datasources/ejb_timer" database="postgresql"
partition="ejb_timer_part" refresh-interval="30000"/>
</data-stores>
</timer-service>
```

以下の表は、サポートされるデータベースと、対応する **database** 属性値を示しています。

表3.1 サポートされているデータベース

データベース	属性値
Hyper SQL (デモを目的としており、サポートはされません)	hsql
PostgreSQL	postgresql
Oracle	oracle
IBM DB2	db2
Microsoft SQL Server	mssql
MySQL および MariaDB	mysql

4. KIE Server および EJB タイマーデータソースを **standalone-full.xml** ファイルに追加します。この例では、**<DATABASE>** はデータベース名、**<SERVER_NAME>** は JNDI データベースのホスト名、**<USER_NAME>** および **<USER_PWD>** はそのデータベースの認証情報になります。

- 以下のように、データソースを追加して、KIE Server がデータベースに接続できるようにします。

```
<xa-datasource jndi-name="java:/jboss/datasources/rhpam" pool-name="rhpam-RHPAM"
use-java-context="true" enabled="true">
  <xa-datasource-property name="DatabaseName"><DATABASE></xa-datasource-
property>
  <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
  <xa-datasource-property name="ServerName"><SERVER_NAME></xa-datasource-
property>
  <driver>postgresql</driver>
```

```

<security>
  <user-name><USER_NAME></user-name>
  <password><USER_PWD></password>
</security>
</xa-datasource>

```

- 以下のように、データソースを追加して、EJB タイマーを有効化します。

```

<xa-datasource jndi-name="java:jboss/datasources/ejb_timer" pool-name="ejb_timer"
use-java-context="true" enabled="true">
  <xa-datasource-property name="DatabaseName"><DATABASE></xa-datasource-
property>
  <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
  <xa-datasource-property name="ServerName"><SERVER_NAME></xa-datasource-
property>
  <driver>postgresql</driver>
  <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  <security>
    <user-name><USER_NAME></user-name>
    <password><USER_PWD></password>
  </security>
</xa-datasource>

```



警告

KIE Server ランタイムデータおよび EJB タイマーデータに、別々のデータベースを使用する必要があります。

5. **<system-properties>** 要素に以下のプロパティを追加し、以下のプレースホルターを置き換えます。

- **<JNDI_NAME>** は、データソースの JNDI 名です。Red Hat Process Automation Manager の場合は **java:/jboss/datasources/rhpam** です。
- **<DIALECT>** は、データベースの hibernate ダイアレクトです。以下の方言がサポートされます。
 - DB2: **org.hibernate.dialect.DB2Dialect**
 - MSSQL: **org.hibernate.dialect.SQLServer2012Dialect**
 - MySQL: **org.hibernate.dialect.MySQL5InnoDBDialect**
 - MariaDB: **org.hibernate.dialect.MySQL5InnoDBDialect**
 - Oracle: **org.hibernate.dialect.Oracle10gDialect**
 - PostgreSQL: **org.hibernate.dialect.PostgreSQL82Dialect**
 - PostgreSQL plus: **org.hibernate.dialect.PostgresPlusDialect**

```
<system-properties>
  <property name="org.kie.server.persistence.ds" value="<JNDI_NAME>"/>
  <property name="org.kie.server.persistence.dialect" value="<DIALECT>"/>
  <property name="org.jbpm.ejb.timer.tx" value="true"/>
</system-properties>
```

6. **standalone-full.xml** ファイルを保存します。
7. クラスターを起動するには **EAP_HOME/bin** に移動して、以下のコマンドの1つを入力します。
 - Linux または UNIX ベースのシステムの場合:

```
$ ./standalone.sh -c standalone-full.xml
```

- Windows の場合:

```
standalone.bat -c standalone-full.xml
```

3.3. ヘッドレス PROCESS AUTOMATION MANAGER コントローラーを使用した KIE SERVER のクラスタリング

Process Automation Manager コントローラーは Business Central と統合します。ただし、Business Central をインストールしない場合は、ヘッドレス Process Automation Manager コントローラーをインストールし、REST API または KIE Server Java Client API を使用してそのコントローラーと対話します。

前提条件

- バックアップを作成してある Red Hat JBoss EAP システム (バージョン 7.3 またはそれ以降) が利用できる。Red Hat JBoss EAP システムのベースディレクトリーを **EAP_HOME** とします。
- インストールを完了するのに必要なユーザーパーミッションが付与されている。
- 「[NFS サーバーの設定](#)」で記載されているように、パーティションをマウントした NFS サーバーが利用できる。

手順

1. Red Hat カスタマーポータル[の「Software Downloads」](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
 - **Product: Process Automation Manager**
 - **バージョン: 7.8**
2. **Red Hat Process Automation Manager 7.8.0 Add Ons**(the **rhpam-7.8.0-add-ons.zip** file) をダウンロードします。
3. **rhpam-7.8.0-add-ons.zip** ファイルを展開します。 **rhpam-7.8.0-controller-ee7.zip** ファイルは展開したディレクトリーにあります。
4. **rhpam-7.8.0-controller-ee7** アーカイブを一時ディレクトリーに展開します。以下の例では、この名前を **TEMP_DIR** とします。

5. **TEMP_DIR/rhpam-7.8.0-controller-ee7/controller.war** ディレクトリーを **EAP_HOME/standalone/deployments/** にコピーします。



警告

コピーするヘッドレス Process Automation Manager コントローラーデプロイメントの名前が、Red Hat JBoss EAP インスタンスの既存デプロイメントと競合しないことを確認します。

6. **TEMP_DIR/rhpam-7.8.0-controller-ee7/SecurityPolicy/** ディレクトリーの内容を **EAP_HOME/bin** にコピーします。ファイルの上書きを確認するメッセージが表示されたら、**Yes** を選択します。
7. **EAP_HOME/standalone/deployments/** ディレクトリーに、**controller.war.dodeploy** という名前で空のファイルを作成します。このファイルにより、サーバーが起動するとヘッドレス Process Automation Manager コントローラーが自動的にデプロイされます。
8. テキストエディターで **EAP_HOME/standalone/configuration/standalone.xml** ファイルを開きます。
9. 以下のプロパティーを **<system-properties>** 要素に追加し、**<NFS_STORAGE>**を、テンプレート設定が保存されている NFS ストレージへの絶対パスに置き換えます。

```
<system-properties>
  <property name="org.kie.server.controller.templatefile.watcher.enabled" value="true"/>
  <property name="org.kie.server.controller.templatefile" value="<NFS_STORAGE>"/>
</system-properties>
```

テンプレートファイルには、特定のデプロイメントシナリオのデフォルト設定が含まれます。

org.kie.server.controller.templatefile.watcher.enabled プロパティーの値を **true** に設定すると、別のスレッドが開始してテンプレートファイルの修正を監視します。この確認の間隔はデフォルトで 30000 ミリ秒になり、**org.kie.server.controller.templatefile.watcher.interval** システムプロパティーで制御できます。このプロパティーの値を **false** に設定すると、テンプレートファイルへの変更の検出が、サーバーの再起動時に制限されます。

10. ヘッドレス Process Automation Manager コントローラーを開始するには、**EAP_HOME/bin** に移動して、以下のコマンドを実行します。

- Linux または UNIX ベースのシステムの場合:

```
$ ./standalone.sh
```

- Windows の場合:

```
standalone.bat
```

第4章 SMART ROUTER のインストールおよび設定

Smart Router (KIE Server ルーター) は、複数の KIE Server、クライアントアプリケーション、他のコンポーネント間の統合レイヤーとして使用可能な軽量の Java コンポーネントです。デプロイメントや実行環境に合わせて、Smart Router は複数の独立した KIE Server インスタンスを単一サーバーのように集約できます。Smart Router には以下の機能があります。

データ集約

クライアントアプリケーションの要求があると、全 KIE Server インスタンス (各グループからインスタンス1つ) からデータを収集し、結果を1つの応答にまとめます。

ルーティング

クライアントアプリケーションからサービスへの呼び出しを受信して、各呼び出しを自動的に個別サービスを実行する KIE Server にルーティングする単一のエンドポイントとして機能します。つまり、KIE Server に同じサービスをデプロイする必要はありません。

負荷分散

負荷分散を効率化します。Smart Router クラスターの負荷分散要求は、標準負荷分散ツールを使用して外部で管理する必要があります。

認証

システムプロパティフラグを使用して KIE Server インスタンスを認証し、HTTPS トラフィックを有効にできます。

環境管理

環境の変更を管理します (例: サーバーインスタンスの追加または削除)

4.1. SMART ROUTER を使用した KIE SERVER インスタンスの負荷分散

Smart Router は、複数の独立した KIE Server インスタンスを集約して1台のサーバーのように使用できます。個々の KIE Server インスタンスに要求をルーティングし、異なる KIE Server インスタンスからデータを集約できるため、インテリジェントなロードバランサーの役割を果たします。エイリアスを介して、Smart Router はプロキシとなります。

前提条件

- 複数の KIE Server インスタンスがインストールされている。



注記

Smart Router を使用するには、KIE Server を管理対象外として設定する必要はありません。

管理対象外の KIE Server はコントローラーに接続されません。

たとえば、管理対象外の KIE Server を Smart Router に接続してコントローラーで Smart Router を登録する場合には、Business Central は Smart Router を使用して管理対象外の KIE Server に問い合わせします。

手順

1. Red Hat カスタマーポータル [「Software Downloads」](#) ページに移動し (ログインが必要)、ドロップダウンオプションから製品およびバージョンを選択します。
 - **Product: Process Automation Manager**

- バージョン: 7.8

2. Red Hat Process Automation Manager 7.8.0 Add-Onsをダウンロードします。
3. ダウンロードした **rhpam-7.8.0-add-ons.zip** ファイルを一時ディレクトリーに展開します。 **rhpam-7.8.0-smart-router.jar** ファイルは展開した **rhpam-7.8.0-add-ons** ディレクトリーにあります。
4. **rhpam-7.8.0-smart-router.jar** ファイルを、ファイルを実行するディレクトリーにコピーします。
5. 以下のコマンドを入力して、Smart Router を起動します。

```
java
-Dorg.kie.server.router.host=<ROUTER_HOST>
-Dorg.kie.server.router.port=<ROUTER_PORT>
-Dorg.kie.server.controller=<CONTROLLER_URL>
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.router.config.watcher.enabled=true
-Dorg.kie.server.router.repo=<NFS_STORAGE>
-jar rhpam-7.8.0-smart-router.jar
```

上のコマンドのプロパティーのデフォルト値は、以下のようになります。

```
org.kie.server.router.host=localhost
org.kie.server.router.port=9000
org.kie.server.controller= N/A
org.kie.server.controller.user=kieserver
org.kie.server.controller.pwd=kieserver1!
org.kie.server.router.repo= <CURRENT_WORKING_DIR>
org.kie.server.router.config.watcher.enabled=false
```

org.kie.server.controller は、以下のように、サーバーコントローラーの URL です。

```
org.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller
```

org.kie.server.router.config.watcher.enabled は、ウォッチャーサービスシステムプロパティーを有効にする任意の設定です。

6. Smart Router に接続する必要のあるすべての KIE Server インスタンスで、 **org.kie.server.router** システムプロパティーを Smart Router の URL に設定します。
7. クライアント側から Smart Router にアクセスするには、以下のように、KIE Server の URL の代わりに Smart Router の URL を使用します。

```
KieServicesConfiguration config =
KieServicesFactory.newRestConfiguration("http://smartrouter.example.com:9000",
"USERNAME", "PASSWORD");
```

この例では、 **smartrouter.example.com** は Smart Router URL で、 **USERNAME** および **PASSWORD** は、 Smart Router 設定のログイン認証情報です。



注記

kie-server に直接コンテナを作成する必要があります。例:

```
$ curl -v -X POST -H 'Content-type: application/xml' -H 'X-KIE-Content-Type: xstream'  
-d @create-container.xml -u ${KIE_CRED} http://${KIE_SERVER_HOST}:${KIE-  
SERVER-PORT}/kie-server/services/rest/server/config/
```

```
$ cat create-container.xml  
<script>  
  <create-container>  
    <container container-id="example:timer-test:1.1">  
      <release-id>  
        <group-id>example</group-id>  
        <artifact-id>timer-test</artifact-id>  
        <version>1.1</version>  
      </release-id>  
      <config-items>  
        <itemName>RuntimeStrategy</itemName>  
        <itemValue>PER_PROCESS_INSTANCE</itemValue>  
        <itemType></itemType>  
      </config-items>  
    </container>  
  </create-container>  
</script>
```

smart-router コンソールにデプロイされたメッセージが表示されます。例:

```
INFO: Added http://localhost:8180/kie-server/services/rest/server as server location for  
container example:timer-test:1.1
```

コンテナの一覧を表示するには、次のコマンドを入力します。

```
$ curl http://localhost:9000/mgmt/list
```

コンテナの一覧が表示されます。

```
{  
  "containerInfo": [{  
    "alias": "timer-test",  
    "containerId": "example:timer-test:1.1",  
    "releaseId": "example:timer-test:1.1"  
  }],  
  "containers": [  
    {"example:timer-test:1.1": ["http://localhost:8180/kie-server/services/rest/server"]},  
    {"timer-test": ["http://localhost:8180/kie-server/services/rest/server"]}  
  ],  
  "servers": [  
    {"kieserver2": []},  
    {"kieserver1": ["http://localhost:8180/kie-server/services/rest/server"]}  
  ]  
}
```

Smart Router の URL を使用してプロセスを初期化するには以下のコマンドを入力します。



```
$ curl -s -X POST -H 'Content-type: application/json' -H 'X-KIE-Content-Type: json' -d
 '{"timerDuration":"9s"}' -u kieserver:kieserver1!
 http://localhost:9000/containers/example:timer-test:1.1/processes/timer-
 test.TimerProcess/instances
```

4.2. TLS 対応の SMART ROUTER の設定

TLS 対応の Smart Router (KIE Server Router) を設定して、HTTPS トラフィックを許可することができます。

前提条件

- Red Hat JBoss EAP 7.3 クラスターの各ノードに KIE Server がインストールされている。
- Smart Router がインストールされ、設定されている。詳細は、[「Smart Router を使用した KIE Server インスタンスの負荷分散」](#) を参照してください。

手順

- TLS サポートと HTTPS を有効にして Smart Router を起動するには、以下の例のように TLS キーストアプロパティを使用します。

```
java -Dorg.kie.server.router.tls.keystore = <KEYSTORE_PATH>
      -Dorg.kie.server.router.tls.keystore.password = <KEYSTORE_PWD>
      -Dorg.kie.server.router.tls.keystore.keyalias = <KEYSTORE_ALIAS>
      -Dorg.kie.server.router.tls.port = <HTTPS_PORT>
      -jar rhpam-7.8.0-smart-router.jar
```

org.kie.server.router.tls.port は、HTTPS ポートの設定に使用されるプロパティです。デフォルトの HTTPS ポート値は **9443** です。

4.3. エンドポイント認証用の SMART ROUTER の設定

エンドポイント認証用に Smart Router (KIE Server ルーター) を設定できます。

前提条件

- Red Hat JBoss EAP 7.3 クラスターの各ノードに KIE Server がインストールされている。
- Smart Router がインストールされ、設定されている。詳細は、[「Smart Router を使用した KIE Server インスタンスの負荷分散」](#) を参照してください。

手順

- エンドポイント認証が有効な Smart Router を起動するには、管理認証情報を設定します。
 - a. 以下のプロパティを KIE Server 設定に追加します。

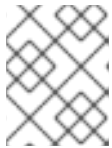
```
`org.kie.server.router.management.username`
`org.kie.server.router.management.password`
```

デフォルトの **username** は KIE Server ID です。

- b. 次のプロパティを Smart Router 設定に追加します。

```
`org.kie.server.router.management.password`
```

password プロパティの値は **true** または **false** (デフォルト) です。



注記

エンドポイント認証を有効にすると、コンテナの表示または追加、削除操作を認証する必要があります。

1. 必要に応じて、Smart Router にユーザーを追加できます。例:

```
java -jar rhpam-7.8.0-smart-router.jar -addUser <USERNAME> <PASSWORD>
```

2. 必要に応じて、Smart Router からユーザーを削除できます。例:

```
java -jar rhpam-7.8.0-smart-router.jar -removeUser <USERNAME>
```

4.4. SMART ROUTER 動作の設定

KIE Server が複数あるクラスター環境では、デフォルトの動作は各 KIE Server に並行して要求を送信し、各 KIE Server のホストには "round-robin" 方式を使用して要求が送信されます。以下の例の環境では、各 KIE Server は同じ KJAR でデプロイされますが、KJAR のバージョンはそれぞれ異なります。

表4.1環境の例

サーバー名	KJAR バージョン	ホスト
kie-server1	kjar:1.0(alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=1.0)	129.0.1.1, 129.0.1.2, 129.0.1.3
kie-server2	kjar:2.0(alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=2.0)	129.0.2.1, 129.0.2.2, 129.0.2.3
kie-server3	kjar:3.0(alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=3.0)	129.0.3.1, 129.0.3.2, 129.0.3.3

要求を送信する場合には、要求が **kie-server1 (129.0.1.2)**、**kie-server2 (129.0.2.3)** および **kie-server3 (129.0.3.1)** に送信されます。

2 番目の要求を送信する場合には、要求は、各 KIE Server の次のホストに送信されます。例: **kie-server1 (129.0.1.3)**、**kie-server2 (129.0.2.1)** および **kie-server3 (129.0.3.2)**。

Smart Router には変更可能なコンポーネントが 3 つあり、この動作を変更できます。

ContainerResolver

サーバーを操作する時に使用するコンテナ ID を検索するコンポーネント

RestrictionPolicy

Smart Router が特定のエンドポイントを使用していないようにするコンポーネント

ConfigRepository

Smart Router 設定を維持するコンポーネント。これは主にルーティングテーブルに関係します。

IdentityService

独自の ID プロバイダーを使用できるようにするコンポーネント。これは KIE Server インスタンス用です。

Smart Router は **ServiceLoader** ユーティリティを使用してこれらのコンポーネントを実装します。

ContainerResolver

META-INF/services/org.kie.server.router.spi.ContainerResolver

RestrictionPolicy

META-INF/services/org.kie.server.router.spi.RestrictionPolicy

ConfigRepository

META-INF/services/org.kie.server.router.spi.ConfigRepository

IdentityService

META-INF/services/org.kie.server.router.identity.IdentityService

たとえば、上記のシナリオでは、Smart Router が利用可能な KIE Server から最新バージョンの KJAR プロセスを検索して、そのプロセスで常に開始するように、**ContainerResolver** をカスタマイズできます。このシナリオでは、各 KIE Server は KJAR を 1 つホストして、バージョンはすべて同じエイリアスを共有します。

Smart Router は実行可能な jar であるため、拡張子を追加するには、コマンドを変更する必要があります。以下に例を示します。

```
java -cp LOCATION/router-ext-7.7.1.redhat-00002.jar:rhcam-7.8.0-smart-router.jar
org.kie.server.router.KieServerRouter
```

サービスが開始されると、コンポーネントに使用されている実装を示すログ出力が表示されます。

```
Mar 01, 2017 1:47:10 PM org.kie.server.router.KieServerRouter <init>
INFO: KIE Server router repository implementation is InMemoryConfigRepository
Mar 01, 2017 1:47:10 PM org.kie.server.router.proxy.KieServerProxyClient <init>
INFO: Using 'LatestVersionContainerResolver' container resolver and restriction policy
'ByPassUserNotAllowedRestrictionPolicy'
Mar 01, 2017 1:47:10 PM org.xnio.Xnio <clinit>
INFO: XNIO version 3.3.6.Final
Mar 01, 2017 1:47:10 PM org.xnio.nio.NioXnio <clinit>
INFO: XNIO NIO Implementation Version 3.3.6.Final
Mar 01, 2017 1:47:11 PM org.kie.server.router.KieServerRouter start
INFO: KieServerRouter started on localhost:9000 at Wed Mar 01 13:47:11 CET 2017
```

第5章 QUARTZ タイマーサービスの設定

クラスターで KIE Server を実行する場合には、Quartz タイマーサービスを設定する必要があります。

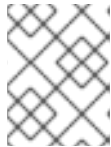
アプリケーションサーバーでデータベースを設定する前に、Quartz テーブルを作成するために Quartz データベースを準備する必要があります。このデータベースで、タイマーデータと Quartz 定義ファイルを保持します。

前提条件

- サポートのある JTA 以外のデータソース (例: PostgreSQL データソースなど) がアプリケーションサーバーに接続されている。

手順

1. データベースに Quartz テーブルを作成し、お使いのデータベース用の DDL スクリプトを使用してタイマーイベントが同期できるようにします。
DDL スクリプトは、**QUARTZ_HOME/docs/dbTables** に展開した、補足用の Zip アーカイブにあります。



注記

quartz_tables_drop_db2.sql など、**drop** の用語が含まれるスクリプトは Quartz テーブルを作成する前に、**drop** が含まれるテーブルを除外します。

2. Quartz 設定ファイル **quartz-definition.properties** を、**JBOSSE_HOME/MODE/configuration/** ディレクトリーに作成して、以下のサンプルコンテンツを追加します。

```
#=====
# Configure Main Scheduler Properties
#=====
org.quartz.scheduler.instanceName = jBPMClusteredScheduler
org.quartz.scheduler.instanceId = AUTO
#=====

# Configure ThreadPool
#=====

org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 5
org.quartz.threadPool.threadPriority = 5
#=====

# Configure JobStore
#=====

org.quartz.jobStore.misfireThreshold = 60000
org.quartz.jobStore.class=org.quartz.impl.jdbcjobstore.JobStoreCMT
org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
org.quartz.jobStore.useProperties=false
org.quartz.jobStore.dataSource=managedDS
org.quartz.jobStore.nonManagedTXDataSource=notManagedDS
```

```

org.quartz.jobStore.tablePrefix=QRTZ_
org.quartz.jobStore.isClustered=true
org.quartz.jobStore.clusterCheckinInterval = 20000
#=====

# Configure Datasources
#=====

org.quartz.dataSource.managedDS.jndiURL=jboss/datasources/psbpmsDS
org.quartz.dataSource.notManagedDS.jndiURL=jboss/datasources/quartzNotManagedDS
# Note the configured data sources that accommodate the two Quartz schemes at the very
end of the file.

```



重要

推奨されるクラスター検出の間隔は 20 秒で、**quartz-definition.properties** ファイルの **org.quartz.jobStore.clusterCheckinInterval** 属性に設定されています。システムへのパフォーマンスの影響を考慮し、必要に応じて設定を変更してください。

3. **org.quartz.properties** プロパティの **quartz-definition.properties** ファイルに絶対パスを指定します。
4. オプション: Quartz トリガーの再試行回数と遅延を設定するには、次のシステムプロパティを更新します。
 - **org.jbpm.timer.quartz.retries** (デフォルト値は 5)
 - **org.jbpm.timer.quartz.delay** (ミリ秒単位、デフォルト値は 1000)



注記

デフォルトでは、Quartz には 2 つのデータソースが必要です。

- プロセスエンジンのトランザクションに参加する管理対象データソース。
- トランザクション処理を行わずにトリガーするタイマーを検索するための管理対象外のデータソース。

Red Hat Process Automation Manager ビジネスアプリケーションでは、Quartz データベース (スキーマ) が Red Hat Process Automation Manager テーブルと共存することを想定しているため、Quartz のトランザクション操作に使用するデータソースを生成します。

他の (トランザクション以外) データソースを設定する必要がありますが、主なデータソースと同じデータベースを参照する必要があります。

第6章 関連資料

- [『Red Hat JBoss EAP 7.3 への Red Hat Process Automation Manager のインストールおよび設定』](#)
- [『Red Hat Process Automation Manager インストールの計画』](#)
- [『Red Hat OpenShift Container Platform への Red Hat Process Automation Manager イミュータブルサーバー環境のデプロイメント』](#)
- [『Red Hat OpenShift Container Platform への Red Hat Process Automation Manager オーサリング環境のデプロイメント』](#)
- [Red Hat OpenShift Container Platform への Red Hat Process Automation Manager フリーフォーム管理サーバー環境のデプロイ](#)
- [『Operator を使用した Red Hat OpenShift Container Platform への Red Hat Process Automation Manager 環境のデプロイメント』](#)

付録A バージョン情報

本書の最終更新日：2021年11月15日（月）